IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
MARSHALL DIVISION

VIRTAMOVE, CORP., *Plaintiff*, v. HEWLETT PACKARD ENTERPRISE COMPANY, *Defendant*.	§ § § § § § § §	CIVIL ACTION NO. 2:24-CV-00093-JRG (LEAD CASE)
VIRTAMOVE, CORP., *Plaintiff*, v. INTERNATIONAL BUSINESS MACHINES CORP., *Defendant*.	§ § § § § § § §	CIVIL ACTION NO. 2:24-CV-00064-JRG (MEMBER CASE)

AMENDED CLAIM CONSTRUCTION ORDER[1]

In these consolidated patent cases, VirtaMove Corp., alleged infringement by Hewlett Packard Enterprise Co. and International Business Machines Corp. (IBM) of claims from U.S. Patents 7,519,814 and 7,784,058. Both patents relate to computer software. *See* '814 Patent at 1:14–16 (noting "[t]he invention relates to computer software" and, more particularly, "management and deployment of server applications"); '058 Patent 1:15–17 (noting the invention "relates to a computing system, and to an architecture that affects and extends services exported through application libraries"). In the member case, IBM counterclaims for infringement by VirtaMove of three related patents[2] that teach application isolation for multiple applications running on a host

[1] The Court issues this Amended Claim Construction Order *sua sponte* solely to amend its construction of a single term relating to "isolated environments." (*See Supra* at 24, 32.) The remainder of the Court's prior Claim Construction Order remains in full force as stated herein.

[2] '634 Patent at [63] (showing the underlying application is a continuation of the applications from

operating system—U.S. Patents 8,943,500, 9,697,038, and 10,606,634—and one patent about cloud computing. *See* '634 Patent at 1:34–39 ("This invention pertains . . . to methods, systems and procedures (i.e., programming) for providing application isolation for multiple applications running on a host operating system."); U.S. Patent 9,722,858 at 1:14–16 ("The present invention relates . . . to cloud computing and the like.").

The Court has since dismissed VirtaMove's claims against Hewlett Packard, Dkt. No. 214, leaving 13 disputes concerning claim scope with IBM. Having considered the parties' briefing, along with arguments of counsel at an April 17, 2025 hearing, the Court resolves those disputes as follows.

I. BACKGROUND

A. U.S. Patent 7,519,814

The '814 Patent "relates to management and deployment of server applications." '814 Patent at 1:15–16. In traditional computer systems, the operating system controls access to shared resources required by various software applications, *see id.* at 1:20–24, but that creates drawbacks that can prevent different applications from being installed on the same system. Most notably, "certain applications require a specific version of operating system facilities and as such will not co-exist with applications that require another version." *Id.* at 1:37–40.

At the time of the application, there were ways to address this. For example, so-called virtual machine technology allowed multiple operating systems "to effectively co-exist on a single compute platform." *Id.* at 1:54–56. That approach, however, "imposes significant performance overhead" and "does nothing to alleviate the requirement that an operating system must be

---

which the '500 Patent and '038 Patent issued).

licensed, managed and maintained for each application." *Id.* at 1:62–65.

The patent purports to address this problem by teaching a system that "only requires one operating system regardless of the number of application containers deployed." *Id.* at 1:60–61. Further, applications can "more effectively share a common comput[ing] platform, and also applications to be easily moved between platforms, without the requirement for a separate and distinct operating system for each application." *Id.* at 1:65–2:3.

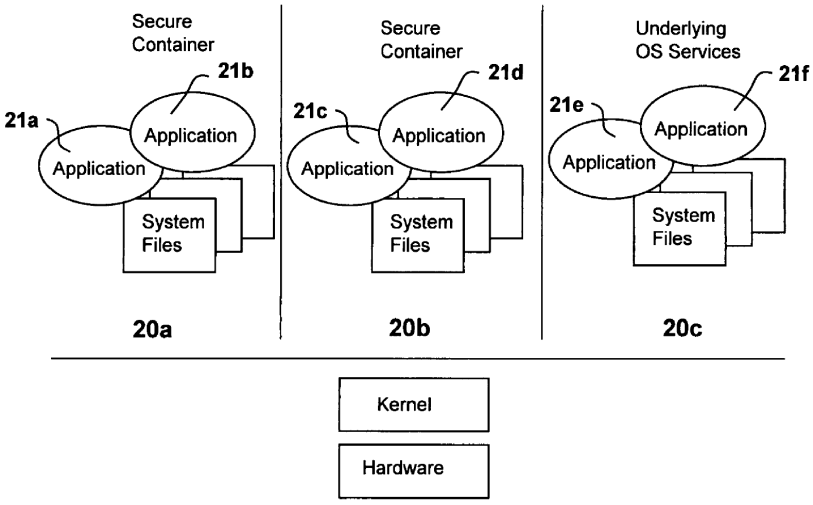

FIG. 2 of the '814 Patent

Figure 2 (above) shows an embodiment with multiple applications (21a–21f). Each application is within one of three secure "containers" (20a–20c), which the server creates by aggregating "all files required to successfully execute a set of software applications on a computing platform." '814 Patent at 7:25–27. Because each application is segregated and executes its own copy of files, each application can use different versions of system files if necessary. *Id.* at 7:8–10.

The parties' disputes concerning this patent relate to Claims 1 and 10. Claim 1 recites:

> 1. In a system having a plurality of servers with operating systems that differ, operating in **disparate computing environments**, wherein each server includes a processor and an operating

> system including a kernel a set of associated local **system files** compatible with the processor, a method of providing at least some of the servers in the system with secure, executable, applications related to a service, wherein the applications are executed in a secure environment, wherein the applications each include an object executable by at least some of the different operating systems for performing a task related to the service, the method comprising:
>
> storing in memory accessible to at least some of the servers a plurality of secure containers of application software, each container comprising one or more of the executable applications and a set of associated **system files** required to execute the one or more applications, for use with a local kernel residing permanently on one of the servers;
>
> wherein the set of associated **system files** are compatible with a local kernel of at least some of the plurality of different operating systems, the containers of application software excluding a kernel,
>
> wherein some or all of the associated **system files** within a container stored in memory are utilized in place of the associated local **system files** that remain resident on the server,
>
> wherein said associated **system files** utilized in place of the associated local **system files** are copies or modified copies of the associated local **system files** that remain resident on the server, and wherein the application software cannot be shared between the plurality of secure containers of application software, and
>
> wherein each of the containers has a unique root file system that is different from an operating system's root file system.

Id. at 17:30–61 (disputed terms in bold). Claim 10, which depends from Claim 1, recites "wherein in operation when an application residing within a container is executed, said application has no access to **system files** or applications in other containers or to system files within the operating system during execution thereof." *Id.* at 18:24–28 (disputed term in bold).

B. U.S. Patent 7,784,058

The '058 Patent relates to a computing architecture "that affects and extends services exported through application libraries." '058 Patent at 1:15–17. The patent explains that, in the prior art, two software applications that require the same critical system element (CSE) might conflict. *Id.* at 1:29–31. Similarly, two software applications that require independent access to specific network services will conflict. *Id.* at 1:31–32.

The prior art addresses these potential conflicts in one of two ways. First, the operating system kernel contains the CSEs, and the applications access those CSEs through application libraries. *See generally id.* at 6:62–7:22. Second, the CSEs "execute in user mode but in a distinct context from applications." *Id.* at 7:24–25. In this case, they are removed from the kernel and "reside in multiple distinct processes or servers." *Id.* at 7:25–27.

The '058 Patent describes "an important distinction" relative to the prior art as "the ability to allow a critical system element (CSE) to execute in the same context as an application." *Id.* at 1:46–48. To provide this capability, the patent teaches deploying multiple instances of the CSE, which allows each application to use a unique instance of the CSE so they can be run simultaneously. *Id.* at 1:48–50. That is, CSEs are not isolated in the kernel, nor are they "removed from the context of an application." *Id.* at 8:64–67. "Rather, they are replicated, and embodied in the context of an application." *Id.* at 8:2–3.

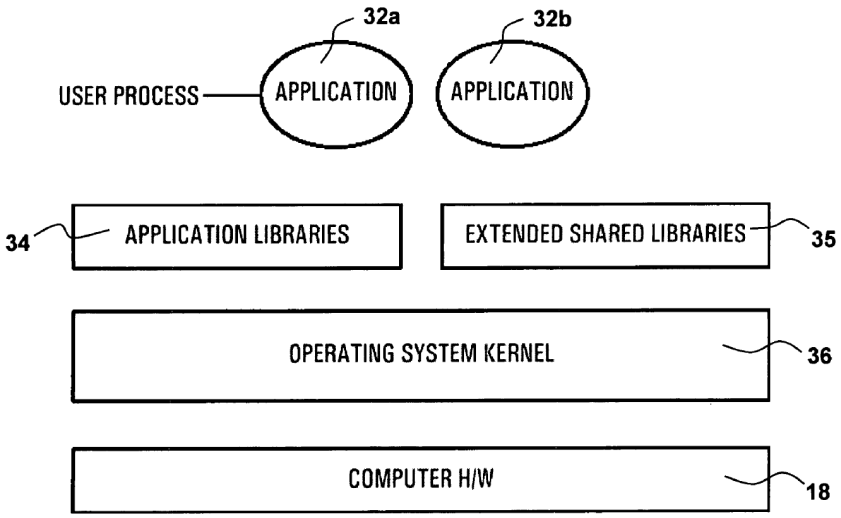

FIG. 3 of the '058 Patent, which shows applications 32 using extended shared libraries 34 in which critical system elements reside.

The claim-construction disputes come from Claim 1, which recites:

1. A computing system for executing a plurality of software applications comprising:

   a) a processor;

   b) an operating system having an operating system kernel having **OS critical system elements** (OSCSEs) for running in kernel mode using said processor; and,

   c) a **shared library** having **shared library critical system elements** (SLCSEs) stored therein for use by the plurality of software applications in user mode and

      i) wherein some of the SLCSEs stored in the **shared library** are **functional replicas** of OSCSEs and are accessible to some of the plurality of software applications and when one of the SLCSEs is accessed by one or more of the plurality of software applications it **forms a part of the one or more of the plurality of software applications**,

      ii) wherein an instance of a SLCSE provided to at least a first of the plurality of software applications from the

> **shared library** is run in a context of said at least first of the plurality of software applications without being shared with other of the plurality of software applications and where at least a second of the plurality of software applications running under the operating system have use of a unique instance of a corresponding critical system element for performing same function, and
>
> iii) wherein a SLCSE related to a predetermined function is provided to the first of the plurality of software applications for running a first instance of the SLCSE, and wherein a SLCSE for performing a same function is provided to the second of the plurality of software applications for running a second instance of the SLCSE simultaneously.

'058 Patent at 10:51–11:14 (disputed terms in bold)

C. U.S. Patents 8,943,500, 9,697,038, and 10,606,634

These related patents, which share the same disclosure, describe systems and methods "for application isolation on host operating systems." '500 Patent at 1:53–55. The patent explains that when applications share system resources, one running application could negatively affect another accessing the same system resources. *Id.* at 1:57–63. The patent addresses this problem by creating an "application isolation environment" provided by an interception layer between the applications and the operating system. *Id.* at 2:57–60. "[A]ny functional changes to systems calls are done exclusively within the interception layer and interception database, and only in context of the calling application." *Id.* at 2:60–63.

Claim 1 of the '500 Patent is exemplary:

> 1. A system, comprising:
>
> one or more central processing units; and
>
> one or more isolated environments including one or more applications and executables;

> wherein the one or more central processing units and the one or more isolated environments are configured to interact with each other;
>
> **wherein the one or more isolated environments are created during installation of the one or more applications**, and updates to the one or more isolated environments occur as the one or more applications use additional resources;
>
> **wherein the one or more isolated environments are removed as part of an uninstall of the one or more applications**;
>
> wherein the one or more isolated environments are stored for retrieval at a later time after the uninstall of the one or more applications.

'500 Patent at 13:23–39 (disputed terms in bold).

The parties have two disputes from these claims. First, VirtaMove challenges Claim 1 of each patent as indefinite because they are directed to a mixed-statutory class. Second, VirtaMove challenges dependent Claim 19 of the '500 Patent and the '038 Patent as indefinite based on lack of antecedent basis.

D. U.S. Patent 9,722,858

The '858 Patent relates to cloud computing. '858 Patent at 1:14–16. It describes Infrastructure as a Service (IaaS), which provides consumers "fundamental computing resources" where the consumer can deploy and run operating systems and applications. *Id.* at 1:20–25. With cloud computing, consumers don't control the underlying infrastructure but have control over operating systems, storage, and deployed applications. *Id.* at 1:25–29.

Most of the disputes from this patent involve Claim 1, which recites:

> 1. A non-transitory computer readable medium comprising computer executable instructions which when executed by a computer cause the computer to perform the method of:
>
> discovering, in a source computing system having a source

> management infrastructure, at least one source infrastructure management component, wherein said at least one source infrastructure management component is an **instance of an image**, and wherein said at least one source infrastructure management component is running in a customer environment;
>
> querying a database to obtain a description of a target cloud infrastructure;
>
> analyzing said at least one source infrastructure management component using said description of said target cloud infrastructure to determine that said at least one source infrastructure management component is **appropriate for infrastructure configuration mapping** to said target cloud infrastructure;
>
> stopping an application executing on said at least one source infrastructure management component determined **appropriate for infrastructure configuration mapping**; and
>
> **capturing** said at least one source infrastructure management component determined **appropriate for infrastructure configuration mapping** for migration to said target cloud infrastructure.

'058 Patent at 10:51–11:14 (disputed terms in bold).

II. LEGAL STANDARDS

A. Generally

"[T]he claims of a patent define the invention to which the patentee is entitled the right to exclude." *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312 (Fed. Cir. 2005) (*en banc*). As such, if the parties dispute the scope of the claims, the court must determine their meaning. *See, e.g.*, *Verizon Servs. Corp. v. Vonage Holdings Corp.*, 503 F.3d 1295, 1317 (Fed. Cir. 2007) (Gajarsa, J., concurring in part); *see also Markman v. Westview Instruments, Inc.*, 517 U.S. 370, 390 (1996), *aff'g,* 52 F.3d 967, 976 (Fed. Cir. 1995) (*en banc*).

Claim construction, however, "is not an obligatory exercise in redundancy." *U.S. Surgical Corp. v. Ethicon, Inc.*, 103 F.3d 1554, 1568 (Fed. Cir. 1997). Rather, "[c]laim construction is a matter of [resolving] disputed meanings and technical scope, to clarify and when necessary to explain what the patentee covered by the claims . . . ." *Id.* A court need not "repeat or restate every claim term in order to comply with the ruling that claim construction is for the court." *Id.*

When construing claims, "[t]here is a heavy presumption that claim terms are to be given their ordinary and customary meaning." *Aventis Pharm. Inc. v. Amino Chems. Ltd.*, 715 F.3d 1363, 1373 (Fed. Cir. 2013) (citing *Phillips,* 415 F.3d at 1312–13). Courts must therefore "look to the words of the claims themselves . . . to define the scope of the patented invention." *Id.* (citations omitted). The "ordinary and customary meaning of a claim term is the meaning that the term would have to a person of ordinary skill in the art in question at the time of the invention, *i.e.,* as of the effective filing date of the patent application." *Phillips*, 415 F.3d at 1313. This "person of ordinary skill in the art is deemed to read the claim term not only in the context of the particular claim in which the disputed term appears, but in the context of the entire patent, including the specification." *Id.*

Intrinsic evidence is the primary resource for claim construction. *See Power-One, Inc. v. Artesyn Techs., Inc.*, 599 F.3d 1343, 1348 (Fed. Cir. 2010) (citing *Phillips*, 415 F.3d at 1312). For certain claim terms, "the ordinary meaning of claim language as understood by a person of skill in the art may be readily apparent even to lay judges, and claim construction in such cases involves little more than the application of the widely accepted meaning of commonly understood words." *Phillips*, 415 F.3d at 1314; *see also Medrad, Inc. v. MRI Devices Corp.*, 401 F.3d 1313, 1319 (Fed. Cir. 2005) ("We cannot look at the ordinary meaning of the term . . . in a vacuum. Rather, we must look at the ordinary meaning in the context of the written description and the prosecution history.").

But for claim terms with less-apparent meanings, courts consider "those sources available to the public that show what a person of skill in the art would have understood disputed claim language to mean . . . [including] the words of the claims themselves, the remainder of the specification, the prosecution history, and extrinsic evidence concerning relevant scientific principles, the meaning of technical terms, and the state of the art." *Phillips*, 415 F.3d at 1314.

B. Indefiniteness

"[A] patent is invalid for indefiniteness if its claims, read in light of the specification delineating the patent, and the prosecution history, fail to inform, with reasonable certainty, those skilled in the art about the scope of the invention." *Nautilus, Inc. v. Biosig Instruments, Inc*., 572 U.S. 898, 901 (2014). The claims "must be precise enough to afford clear notice of what is claimed" while recognizing that "some modicum of uncertainty" is inherent due to the limitations of language. *Id.* at 909. "Indefiniteness must be proven by clear and convincing evidence." *Sonix Tech. Co. v. Publ'ns Int'l, Ltd.*, 844 F.3d 1370, 1377 (Fed. Cir. 2017).

III. THE LEVEL OF ORDINARY SKILL IN THE ART

The level of ordinary skill in the art is the skill level of a hypothetical person who is presumed to have known the relevant art at the time of the invention. *In re GPAC*, 57 F.3d 1573, 1579 (Fed. Cir. 1995). In resolving the appropriate level of ordinary skill, courts consider the types of and solutions to problems encountered in the art, the speed of innovation, the sophistication of the technology, and the education of workers active in the field. *Id.* Importantly, "[a] person of ordinary skill is also a person of ordinary creativity, not an automaton." *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 421 (2007).

Here, neither VirtaMove nor IBM propose a level of ordinary skill in the art in their briefing. IBM's expert, however, characterizes a skilled artisan as one who "would have a bachelor's

degree in computer science or a similar degree, and at least two years of experience in the field of computer software systems." Stavrou Decl., Dkt. No. 143-7 ¶ 16. Because that characterization is unrebutted, the Court adopts it for analysis but limits the amount of experience to two years.

IV. THE DISPUTED TERMS

A. "disparate computing environments" ('814 Patent, Claim 1)

VirtaMove's Construction	IBM's Construction
"environments run by standalone computers" Alternatively, "environments on standalone computers or on computers that are unrelated" (Dkt. No. 156 at 1)	"environments where computers are standalone or where there are multiple computers and where they are unrelated"

This term appears in the preamble of Claim 1, which the parties agree is limiting:

> 1. In a system having a plurality of servers with operating systems that differ, operating in *disparate computing environments*, wherein each server includes a processor and an operating system including a kernel a set of associated local system files compatible with the processor, a method of providing at least some of the servers in the system with secure, executable, applications related to a service . . . .

'814 Patent at 17:30–36 (emphasis added). The parties dispute whether the required "disparate computing environments" are *on* computers or whether they *contain* computers. IBM contends the patent's lexicography requires the latter.

VirtaMove says "the specification makes clear that computing environments are disparate 'where computers are stand-alone or where there are plural computers and where they are unrelated.'" Dkt. No. 143 at 1 (quoting '814 Patent at 2:17–19). VirtaMove characterizes IBM's construction as "touch[ing] on these similar principles," but objects to what it calls "absurd results" when that construction is substituted for the claim language. *Id.* at 2. It its view, its construction

"clarifies" that "environment" is not an environment like a house or a city, but a computing environment. *Id.* at 3.

IBM asserts lexicography and quotes the same definitional language on which VirtaMove relies. Dkt. No. 151 at 3 (citing '814 Patent at 2:17–19). It then criticizes VirtaMove's approach as altering one part of the definition provided by the specification and omitting the other. It says "VirtaMove's injection of the new phrase 'run by' fundamentally changes the claim's meaning by introducing an operational requirement—suggesting the term is met by a single stand-alone computer simply running an environment rather than multiple computers existing within distinct environments." *Id.* at 4.

In its reply, VirtaMove says it "agrees with [IBM as to] what the concept of 'disparate computing environments' is getting at." Dkt. No. 156 at 2 n.1. It also proposes an alternative construction that replaces "run by" with "on," thereby more closely aligning its construction with IBM's. Its concern is the use of "where" in IBM's construction, which it calls "ambiguous." *Id.*

The Court agrees with VirtaMove. Putting aside the three-line "definition" for now, nothing in the patent suggests the invention in any way relates to where computers are. To the contrary, the Background of the Invention describes the problem of applications that require different operating systems to function. *See generally* '814 Patent at 1:20–50. To address that problem, "the invention . . . offers the ability for applications to more effectively share a common compute platform, and also allow applications to be more easily moved between platforms, without the requirement for a separate and distinct operating system for each application." *Id.* at 1:65–2:3; *see also id.* at 2:9–11 (noting one software product "does not isolate applications into distinct environments," and "[a]pplications executing within [its] environment don't possess a unique identity"). *id.* at 4:3–6 ("the invention provides a method of establishing a secure environment for executing, on a

computer system, a plurality of applications that require shared resources").

The "definition" does nothing to change this understanding. Those three lines define "[d]isparate computing environments" as "[e]nvironments where computers are stand-alone or where there are plural computers and where they are unrelated." '814 Patent at 2:17–19. This definition could reasonably be read by a skilled artisan to mean that unrelated computers *per se* have different environments, which would be consistent with the rest of the specification. In fact, IBM's briefing even suggests as much, pointing, for example, to Figure 1 as an example of two unrelated servers each with its own hardware and operating system. Dkt. No. 151 at 4. Notably, Figure 1 and its related description don't speak to where the two computers are.[3] Also, IBM focuses on use of the word "where," but the patent elsewhere uses that same word synonymously with "in which." *See e.g.*, *id.* at 2:43–48 (defining "secure application container" as "[a]n environment where [i.e., in which] each application set appears to have individual control of some critical system resources and/or where [i.e., in which] data within each application set is insulated from the effects of other application sets").

The Court concludes, based on the specification as a whole, that the three lines on which IBM relies do not clearly redefine this term to limit where the computers of the claimed system must be located. Accordingly, the Court construes this term as "environments on standalone computers or on computers that are unrelated."

[3] In fact, all the citations from IBM's brief use the word "environment" as something on a computer rather than where the computer is located. *See* Dkt. No. 151 at 4 (noting the specification "describes applications ***in*** an environment ***on*** a computer" and citing '814 Patent at 1:27–30 ("environment where a collection of applications . . . must be separated with each application installed on an individual computer"), 7:5–6 (noting applications can "execute in a secure environment").

B. "system files" ('814 Patent, Claim 1)

VirtaMove's Construction	IBM's Construction
No construction necessary; plain and ordinary meaning "files provided with an operating system and which are available to applications as shared libraries and/or configuration files" (Dkt. No. 156 at 4)	"files provided within an operating system and which are available to applications as shared libraries and configuration files"

"System files" appears seven times in Claim 1. VirtaMove says this term has an ordinary meaning of "operating system files that include (but are not limited to) shared library files and configuration files." Dkt. No. 143 at 3. VirtaMove accuses IBM of "introduc[ing] ambiguity by defining 'system files' as a collective, rather than defining what individual system files are." *Id.* at 4. It says IBM's construction wrongly "requir[es] each (individual) system file [to] meet the requirement of the proposed construction[,] which addresses 'system files' (collectively)." *Id.* Moreover, says VirtaMove, IBM's construction wrongly implies "system files" are limited to only "shared library" and "configuration files." *Id.*

In response, IBM points to an express definition of the term and characterizes VirtaMove's arguments opposing that lexicography as "straw men." Dkt. No. 151 at 7. It denies that each system file must be both a shared library and a configuration file, but it then suggests the phrase "system files" requires both types of files. *Id.* at 7–8 ("the express definition of system *files* (also plural) requires both "shared libraries *and* configuration files"). In other words, multiple shared-library files or multiple configuration files are not by themselves enough to meet this limitation. IBM emphasizes the specification's example of Linux Apache's shared libraries and configuration files as "system" files, which appears right after the definition on which IBM relies. *Id.* at 8 (quoting '814 Patent at 2:55–3:16).

The Court agrees with VirtaMove. A skilled artisan could reasonably read the three-line definition to mean system files are any combination of multiple shared libraries and configuration files. In fact, the Linux Apache example supports that by referring only to shared libraries as "system files." '814 Patent at 4:55–57 ("Linux Apache uses the following shared libraries, supplied by the OS distribution, which are 'system' files."). Thus, the specification's definition of "system files," especially when considered with the Apache example, does not so clearly and unambiguously evidence an intent to define "system files" (plural) to require both types of files. If one "system file" can be either a shared library or a configuration file, then "system files" are simply more than one "system file." Accordingly, the Court adopts VirtaMove's position and construes "system files" as "files provided within an operating system and which are available to applications as shared libraries and/or configuration files."

C. "critical system elements" / "operating system critical system elements" / "shared library critical system elements" ('058 Patent, Claim 1)

VirtaMove's Construction	IBM's Construction
"critical system elements": "any service or part of a service, 'normally' supplied by an operating system, that is critical to the operation of a software application"	Indefinite

Claim 1 recites both "operating system critical system elements (OSCSEs)" and "shared library critical system elements (SLCSEs)." '058 Patent at 10:54–59. IBM says "critical system element" is an inherently subjective term because it relates to degrees of importance, and "neither the intrinsic record nor the relevant art provides any objective standard to guide a [skilled artisan] in determining whether a service is 'critical' to the operation of a software application." Dkt. No. 151 at 9. IBM recognizes the specification's purported definition of the term, but calls it circular. *Id.* at 10. Moreover, IBM's expert cites a technical dictionary definition of "criticality" that calls it

"subjective." *Id.* at 11 (citing Stavrou Decl., Dkt. No. 151-1 ¶ 33). And it cites its expert's opinion that a skilled artisan "could select from any number of considerations to determine criticality." *Id.* (citing Stavrou Decl., Dkt. No. 142-7 ¶¶ 34–38, 46–47).

VirtaMove points to the patent's definition and argues the specification provides "further context." Dkt. No. 143 at 5. It criticizes IBM's expert for ignoring the claim language. *Id.* at 6. In fact, says VirtaMove, "that criticality of services would be understandable to a POSITA given the context of both the operating system and software applications expressly undermines [IBM's] indefiniteness contention." *Id.* at 7; *see also* Dkt. No. 156 at 5 (asserting IBM's expert "did not specify what his conclusion on indefiniteness would be under a scenario where a POSITA did know the specific software application [and] operating system in question").

This term is not indefinite. The patent defines "Critical System Element (CSE)" as "[a]ny service or part of a service, 'normally' supplied by an operating system, that is critical to the operation of a software application." '058 Patent at 6:6–8. That is, admittedly, not helpful to the meaning of "critical," but the patent continues: "A CSE is a dynamic object providing some function that is executing instructions used by applications." *Id.* at 6:9–10. *That* is the criticality, because the application won't work as intended without it. The breadth of the examples provided by the specification supports that conclusion. *See id.* at 6:12–28 (referring to network services, message passing protocols, file-system services, file-system optimizations, and network optimizations). Accordingly, the Court construes this term as "a dynamic object providing some function that is executing instructions used by applications."

D. "functional replicas" ('058 Patent, Claim 1)

VirtaMove's Construction	IBM's Construction
"substantial functional equivalents or replacements of kernel functions"	Indefinite

The claim recites "a shared library having shared library critical system elements (SLCSEs) stored therein for use by the plurality of software applications in user mode and i) wherein some of the SLCSEs stored in the shared library are *functional replicas* of OSCSEs." '058 Patent at 10:57–61 (emphasis added). IBM challenges "functional replicas" as indefinite because the specification allows for a not-quite-exact copy, but fails to explain how similar of a copy is similar enough. Dkt. No. 151 at 15.

In its reply, VirtaMove points to the specification's definition of "replica" that aligns with its construction and asserts that it conveys a "definite scope" about the term's meaning. Dkt. No. 156 at 6. Suggesting a doctrine-of-equivalents-like analysis is appropriate, VirtaMove further emphasizes that juries regularly determine equivalency, including whether they perform "substantially the same function." *Id.* at 7. Thus, says VirtaMove, IBM's "demand for additional 'objective guidance' . . . contradicts established law." *Id.* At the hearing, VirtaMove's main criticism of IBM's position was that its expert did not consider that a factfinder would understand whether something is substantially equivalent in function to something else. Hr'g Tr., Dkt. No. 216 at 55:13–18.

The Court agrees with IBM. While the ordinary meaning of "replica" might be definite, the specification *defines* "replica" to mean something *less than* an exact copy. Specifically, the patent explains "[t]he term replica used herein is meant to denote a CSE having similar attributes to, but not necessarily, and preferably not an exact copy of a CSE in the operating system." '058

Patent at 1:66–2:1. But nowhere does the patent explain how close the replica must be, either in structure or function, to the CSEs. In fact, VirtaMove's construction highlights the problem by including a term of degree.

VirtaMove's briefing does little to address the uncertainty about the term's scope. Instead, VirtaMove emphasizes "functional replica" has a broader scope than "replica." That is true, but that does not impose any objective boundaries on the scope of the term to make a definite to a skilled artisan, particularly considering the patent's definition of the term. VirtaMove's attempt to conflate a doctrine-of-equivalents analysis with an indefiniteness determination is also not persuasive, because the former first requires definite structure before equivalents to that structure can be determined.

IBM has shown that a skilled artisan would not be reasonably certain about the scope of "functional replicas." Accordingly, the Court holds this term is indefinite.

E. "forms a part of the one or more of the plurality of software applications" ('058 Patent, Claim 1)

VirtaMove's Construction	IBM's Construction
No construction necessary; plain and ordinary meaning	"literally form a part of the application such that it resides in the same address space as application code, in contrast to a proxy that is exclusive of the application"

The relevant limitation recites "when one of the SLCSEs is accessed by one or more of the plurality of software applications it *forms a part of the one or more of the plurality of software applications*." '058 Patent at 10:62–65. During prosecution, the patent examiner cited O'Rourke as disclosing this limitation. In response, the applicants characterized O'Rourke's "controlling agent" as querying the SLCSEs for information to "interconnect kernel mode filters to create a

filter graph." '058 Patent File History, Dkt. No. 143-8 at 33. "Indeed," the applicants explained,

> the controlling agent 44 fails to disclose that when one of the SLCSEs is accessed by one or more of the plurality of software applications, it forms a part of the one or more of the plurality of software applications. In other words, *the SLCSEs literally form part of the application. SLCSEs reside in the same address space as application code, in contrast to a proxy that is exclusive of the application.*

Id. (emphasis added).

There is not much disagreement on the meaning of the phrase. VirtaMove's objection to IBM's construction focuses on the word "literally," which it says gives undue emphasis to this limitation. Dkt. No. 143 at 11. In its reply, VirtaMove objects to considering the prosecution history language definitional "because not everything that 'resides in the same address space as' the application's application code 'form[s] a part of the application.'" Dkt. No. 156 at 8. VirtaMove agrees, however, that for a system element to "form a part of the one or more software applications," it must "reside in the same address space as the application code." Dkt. No. 143 at 10.

The Court agrees with VirtaMove that "literally" in IBM's construction could be confusing. The prosecution history explains what the phrase means, which is that the "SLCSEs reside in the same address space as the application code." But the phrase "in contrast to . . ." is simply a way of saying "in contrast to SCLEs that *don't reside* in the same address space as application code," and thus adds no value. Accordingly, the Court construes this phrase as "resides in the same address space as application code of the one or more of the plurality of software applications."

F. "shared library" ('058 Patent, Claim 1)

VirtaMove's Construction	IBM's Construction
"an application library whose code space is shared among all user mode applications"	"an application library code space shared among all user mode applications. The code space is different than that occupied by the kernel and its associated files. The shared library files are placed in an address space that is accessible to multiple applications," wherein an "application library" is "a collection of functions in an archive format that is combined with an application to export system elements"

This disputes centers on a purported definition for "shared library," which Claim 1 requires to store "shared library critical system elements (SLCSEs) . . . for use by the plurality of software applications in user mode." '058 Patent at 10:57–59. According to IBM, the patent defines "shared library" as "[a]n application library code space shared among all user mode applications." *Id.* at 6:49–50. Based on their respective constructions, the parties dispute whether a shared library *has* a code space or whether it *is* a code space.

VirtaMove calls IBM's construction confusing and circular. It suggests the specification's "definition" is wrong and stems from editorial or typographical errors, and cites a related provisional application's definition of "[a]n application library whose code space is shared among all user applications." Dkt. No. 143 at 12 (quoting Provisional Appl'n 60/504,213, Dkt. No. 143-9 at 9). According to VirtaMove, when the applicants drafted the non-provisional application, they "want[ed] to include some clarification in the second and third sentences and, in the course of that, kind of made a hash of the first sentence." Hr'g Tr., Dkt. No. 216 at 58:4–7; *see* Dkt. No. 143 at 13 ("the correct definition without the typographical errors should be included").

IBM relies on the first sentence of the definition in the specification, and then argues the claims and specification support that definition. Regarding the provisional application, IBM says

its content materially differs from the patent's specification, and thus should have little weight. Dkt. No. 151 at 20. VirtaMove's construction, says IBM, fundamentally alters the claim scope by ignoring that the "shared library" has a specific type of memory space for storing different things. *Id.* at 21.

The Court agrees with VirtaMove on the first part of IBM's construction. First, although IBM says the provisional application is materially different from the actual patent, it provides no analysis of that difference. Dkt. No. 151 at 20. Second, the specification provides a definition for "static library," '058 Patent at 6:54–55 ("Static library: An application *whose* code space is *contained* in a single application"), and a skilled artisan would expect the definitions for "shared library" and "static library" to align at least as to the meaning of "library." In other words, a skilled artisan would not understand a "shared library" to be the code space, but the "static library" to *not* be the code space, when the only difference between those phrases is "shared" or "static"—especially given the different definition provided by provisional application.

Finally, the specification describes a "shared library" as *having* a code space rather than being code space. *See* '058 Patent at 7:3–5 ("[W]hat is commonly done is to provide an application library *in* shared code space, which multiple applications can access." (emphasis added)). Thus, the specification as a whole does not clearly and unambiguously redefine the meaning of "shared library" to be "code space."

That said, the Court agrees with IBM that the second sentence of its construction should be included. VirtaMove does not contest this. *See* Dkt. No. 156 at 9 ("VirtaMove does not believe that including or excluding this sentence substantively affects claim scope."). As for the third sentence of IBM's construction, which nests an "application library" definition from the specification into the "shared library" construction, the parties have not presented a dispute about the meaning

of "application library." The Court therefore construes "shared library" as "an application library whose code space is (1) shared among all user mode applications, and (2) different than that occupied by the kernel and its associated files."

G. "A system, comprising . . . wherein the one or more isolated environments are created during installation of the one or more applications . . . wherein the one or more isolated environments are removed as part of an uninstall of the one or more applications" ('500, '634, and '038 Patents, Claim 1)

IBM's Construction	VirtaMove's Construction
Plain and ordinary meaning	Indefinite. Alternatively, requiring the "creating" and "removing" to be performed "independently of user intervention.

VirtaMove, as an infringement counter-defendant, challenges these claims as indefinite because they cover both a method of use and an apparatus. Dkt. No. 152 at 1. VirtaMove's challenge is based on the preambles, which recite "a system," and the two "wherein" clauses that reference creation and removal of "one or more isolated environments." *Id.* at 2. VirtaMove cites IBM's expert, who testified that "isolated environments must be created during one or more of the installations" for the claims to be practiced. *Id.* at 3 (citing Stavrou Depo. Tr., Dkt. No. 152-2 at 9:7–10:15).

IBM counters that the "wherein" clauses are capabilities of the system. Dkt. No. 142 at 6. It distinguishes the cases on which VirtaMove relies as claiming a user action rather than a capability of the system. *Id.* at 8. It also accuses VirtaMove of mischaracterizing IBM's expert's testimony. Dkt. No. 153 at 3.

The Court agrees with IBM. VirtaMove, as the challenging party, must prove indefiniteness by clear and convincing evidence. *Sonix Tech. Co.*, 844 F.3d at 1377. As such, if a skilled artisan can reasonably read the claim language in a way consistent with only one statutory class, that weighs against an "indefiniteness" holding.

VirtaMove stresses the lack of "configured to" in these claims, and that when the applicant wanted to claim the system was "configured to" perform a step, the patentee used that phrase. Dkt. No. 152 at 3 (citing the limitation "wherein the one or more [CPUs] and the one or more isolated environments are configured to interact with each other"). That, however, reads "configured to" in isolation from the remaining claim language. The next two limitations—the "wherein" clauses at issue—limit the nature of the configuration.[4] In other words, the two "wherein" clauses explain the required "interaction." Thus, a skilled artisan would be reasonably certain the challenged language requires the system *to be configured* to (1) to create isolated environments during installation of the applications, and (2) remove the isolated environments as part of an uninstall.

As an alternative, VirtaMove urges the Court to construe these terms to be performed "independently of user interaction." For support, VirtaMove points to what it calls IBM's "explicit representations" of such a requirement. Dkt. No. 152 at 10. But IBM represented only that "the claims are agnostic as to how these actions are triggered, whether it is via an automated process or some unclaimed user input." Dkt. No. 142 at 8. That is, IBM merely said the specification's description of the system's operation "does not require any user action or intervention," *id.* at 9, not that it must be a requirement of the claims. Accordingly, the Court rejects VirtaMove's alternative position and construes (1) "wherein the one or more isolated environments are created during installation of the one or more applications" as "wherein the system is configured to create the one or more isolated environments during installation of the one or more applications" and (2) "wherein the one or more isolated environments are removed as part of an uninstall of the one or more applications" as "wherein the system is configured to remove the one or more isolated environments as part of an uninstall of the one or more applications."

[4] So does the last "wherein" clause, which is not at issue for claim construction.

H. "the system resources" ('500 Patent, Claim 19; '038 Patent, Claim 19)

IBM's Construction	VirtaMove's Construction
Plain and ordinary meaning	Indefinite

Claim 19 of each patent recites, in its entirety, "[t]he non-transitory computer readable storage medium of claim 18 comprising instructions for maintaining mapping between the system resources inside the one or more isolated environments and outside." '500 Patent at 14:50–53; *see also* '038 Patent at 15:5–8. The claims from which they depend, however, do not recite "system resources." VirtaMove challenges this claim as indefinite because "the system resources" has no antecedent basis.

IBM says the claim language itself provides the requisite clarity. In its view, a skilled artisan "would understand that the claimed 'system resources' can be met by any system resources that satisfy the recited description." Dkt. No. 142 at 11; *see also* Dkt. No. 153 at 5 ("[T]his term's plural form can refer to mapping one or more system resources inside the isolated environment with one or more system resources outside the isolated environment."). But VirtaMove says that's precisely the issue, asking "*which* system resources satisfy the requirements of Claim 18 and Claim 19." Dkt. No. 152 at 14.

The claims are not indefinite. A skilled artisan, or for that matter a lay person, would not read "the system resources" within an environment to be only *some* of the system resources within the environment. By referring to "*the* system resources inside the isolated environment," that means "*all* the system resources inside the environment"—not just those that happen to be mapped to the outside. By analogy, "the rooms inside the house" refer to *all* rooms inside the house, not just *some* of the rooms inside the house. The Court therefore construes this phrase as "all of the system resources."

I. "appropriate for infrastructure configuration mapping" ('858 Patent, Claims 1, 19)

IBM's Construction	VirtaMove's Construction
Plain and ordinary meaning, which requires an objective determination.	Plain and ordinary meaning

This dispute started as an "indefiniteness" challenge by VirtaMove contending "appropriate" is a subjective term of degree. In its opening brief, IBM asserted that, "in the context of the infrastructure configuration mapping required by the claim language, . . . this term recites an objective determination that is therefore not indefinite." Dkt. No. 142 at 14. VirtaMove responded by withdrawing its indefiniteness challenge but requesting a construction that requires an objective determination based on representations made by IBM. Dkt. No. 152 at 15. IBM replies this would be inappropriate because VirtaMove provides no support or explanation as to what the determination entails. Dkt. No. 153 at 6.

The Court agrees with VirtaMove. It its opening brief, IBM emphasizes the "objective" nature of the determination. *See* Dkt. No. 142 at 16 (concluding "the intrinsic record provides a sufficient description and examples of the required objective analysis" to understand the term with reasonable certainty and that "[e]xtrinsic evidence also supports that the term recites an objective determination"). Accordingly, the Court clarifies that the ordinary meaning of this requires an objective determination of "appropriateness."

J. "instance of an image" ('858 Patent, Claims 1, 19)

IBM's Construction	VirtaMove's Construction
Plain and ordinary meaning, which is "an occurrence or copy of an image"	Indefinite. Alternatively: "Image": "a template that includes virtual hardware suggestions and a virtual disk containing at least an operating system" "Instance of an image": a virtual machine derived from an image, which further includes virtual hardware allocations and a hypervisor of virtual machine runtime.

This dispute concerns two terms—"instance" and "image." VirtaMove asserts lexicography based on the following definitions from the patent:

- "Instance: An operating system instance together with all software running on this operating system. It may be physical (i.e., directly running on a server) or virtual (i.e., already running on a hypervisor)." '858 Patent at 15:30–33.
- "Source instance: Instance as it is running on the source side, before migration." *Id.* at 15:34–35.
- "Image: File representation of an instance." *Id.* at 15:36.
- "Catalog image: Image in a cloud catalog, to be used if new instances are created in the cloud from scratch, rather than by rapid migration." *Id.* at 15:37–39.

VirtaMove, however, says this lexicography is circular, unhelpful, inconsistent, and therefore indefinite. Dkt. No. 152 at 16–17. As an alternative, VirtaMove points to a different part of the specification, which it says avoids the confusion of these "definitions." *Id.* at 17–18 (citing '858 Patent at 50:52–51:10). IBM accuses VirtaMove of importing limitations from some embodiments and excluding others. Dkt. No. 153 at 7.

The patent's "lexicography" looks recursive at first, but the specification shows why. The specification defines an "image" as a "file representation of an instance," and the disputed term is "instance of an image." This seems tautological, but the specification discloses capturing images

of "instances" from a machine and then, if needed, recovering the state of the machine by instantiating a new instance from the image. *See* '858 Patent at 46:55–60 ("when a rollback is desired, snapshot management system checks the image out of [a] reference repository as a template, and places it into [an] operational repository . . . and then reconfigures the template to match an existing VM instance and instantiates a new virtual machine based on the template" (reference numbers omitted)). In other words, the new instance is intended to be as close to the old instance as possible, but they are different instances. Thus, a better way to understand the disputed phrase is "instance of an image of a previous (or different) instance." The term is not indefinite.

K. "capturing" ('858 Patent, Claim 1)

IBM's Construction	VirtaMove's Construction
Plain and ordinary meaning	Plain and ordinary meaning, which is "transferring into a file"

The relevant limitation recites "capturing said at least one source infrastructure management component . . . for migration to said target cloud infrastructure." '858 Patent at 71:45–48. VirtaMove initially contended this term is indefinite, but has withdrawn that challenge. Dkt. No. 152 at 22. VirtaMove now seeks to exclude from the term's scope "recording, acquiring, or extracting data, inputs, or events from a system, environment, or stream." *Id.* at 23.

Ultimately, the parties dispute whether the result of the "capturing" step must be a file. In its briefing VirtaMove said the "capturing" is for a specific purpose—"migration to said target cloud infrastructure"—and that requires first writing to a file. Dkt. No. 152 at 23. But VirtaMove softened that position at the hearing based on comments from IBM. Hr'g Tr., Dkt. No. 216 at 94:4–7 (suggesting only that "the thing captured must be appropriate for migration to said target cloud infrastructure"). Under IBM's construction, says VirtaMove, a remote desktop application that

monitors user input and then transmits that input to the cloud infrastructure would fall within the scope of the claims. Dkt. No. 152 at 23.

In contrast, IBM says data can be "captured" and then "transported to [a] cloud location." Dkt. No. 153 at 9. Regarding VirtaMove's concern about user input, IBM stresses "[t]hat is simply not the way these claims are read." Hr'g Tr., Dkt. No. 216 at 92:18. Rather, the limitation requires capturing a "source infrastructure management component" (not user input), which "must somehow allow for migration." *Id.* at 92:20–93:1.

The Court sees no reason the ordinary meaning of "capturing" requires writing to a file. VirtaMove's argument seems geared to the *practicality* of an embodiment that does not write captured data to a file. But whether an embodiment of a claim is practical does not determine the claim's scope. The Court therefore rejects VirtaMove's construction as outside the scope of the ordinary meaning of the term.

L. "non-functional requirement" ('858 Patent, Claims 3–5, 7)

IBM's Construction	VirtaMove's Construction
Plain and ordinary meaning	Indefinite. Alternatively, "requirements unrelated to the functionality of a system"

VirtaMove's challenge stems partly from IBM's failure to propose a construction for the term. Dkt. No. 152 at 26 ("[G]iven IBM's inability to articulate any coherent (even high-level) meaning of what non-functional requirements are, the term should be held indefinite."). It also focuses on the specification's disclosure of a service level agreement (SLA). *Id.* The specification makes clear, says VirtaMove, that service level agreements, or "SLAs," may or may not be "non-functional requirements," and the intrinsic record provides no clarity as to which portions of an SLA would or would not constitute non-functional requirements. *Id.*

This, however, does not show the term is indefinite. VirtaMove has that burden, and here IBM's failure to propose a construction, without more, is not sufficient to meet it. Notably, the Court is not construing "service level agreement" or "SLA," nor have the parties briefed that term. And although VirtaMove criticizes IBM for failing to "provide clarity as to which portions of an SLA would or would not constitute non-functional requirements," Dkt. No. 152 at 26, VirtaMove was free to undertake that task in support of its position and chose not to.

Alternatively, VirtaMove asks the Court to construe this phrase as "'requirements unrelated to the functionality of a system' so that IBM is unable to take advantage of the obvious ambiguity of the intrinsic record." Dkt. No. 152 at 26 n.5. IBM does not address this alternative construction, but the Court struggles to see why it should substitute what even VirtaMove calls a "close paraphrasing" for the disputed term, and how that would address what VirtaMove considers to be the problem. The Court therefore declines VirtaMove's request and will give this term a "plain and ordinary meaning" construction.

M. "cloud infrastructure" ('858 Patent, Claims 1, 4–6, 8–12, 19)

IBM's Construction	VirtaMove's Construction
Plain and ordinary meaning, which is "a network of interconnected nodes."	Plain and ordinary meaning, which is: an infrastructure comprising a network of interconnected nodes that provides for on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service providing transparency for both the provider and consumer of the utilized service.

The parties dispute the ordinary meaning of the term. In IBM's view, VirtaMove proposes a "needlessly restrictive" construction. Dkt. No. 142 at 29. IBM bases its construction on a quote from the patent, which its expert opines is the terms ordinary meaning. *Id.* at 30 (quoting '858

Patent at 5:29–33). VirtaMove counters that, under IBM's construction, "every network of interconnected nodes is *ipso facto* a 'cloud architecture.'" Dkt. No. 152 at 27. It stresses the patent's citation to a NIST document that defines the term, and points to extrinsic evidence published by IBM that identifies "five essential characteristics of a clouds" that align with its construction. *Id.* at 28.

The Court agrees with VirtaMove. To be sure, "[a]t the heart of cloud computing is an infrastructure comprising a network of interconnected nodes." '858 Patent at 5:31–33. But the extrinsic evidence shows a skilled artisan would understand "cloud computing," and thus the infrastructure needed to support it, to be more than just "interconnected nodes." Based on the NIST definition of "cloud computing" and "cloud architecture," the Court construes "cloud infrastructure" as "collection of hardware and software that enables on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service." *See The NIST Definition of Cloud Computing*, Dkt. No. 152-4 at 2 n.2 ("A cloud infrastructure is the collection of hardware and software that enables the five essential characteristics of cloud computing.").

V. CONCLUSION

Disputed Term	The Court's Construction
"disparate computing environments" ('814 Patent, Claim 1)	"environments on standalone computers or on computers that are unrelated"
"system files" ('814 Patent, Claim 1)	"files provided within an operating system and which are available to applications as shared libraries and/or configuration files"
"critical system elements" / "operating system critical system elements" / "shared library critical system elements" ('058 Patent, Claim 1)	**"critical system element":** "a dynamic object providing some function that is executing instructions used by applications"

"functional replicas" ('058 Patent, Claim 1)	Indefinite
"forms a part of the one or more of the plurality of software application" ('058 Patent, Claim 1)	"resides in the same address space as application code of the one or more of the plurality of software applications"
"shared library" ('058 Patent, Claim 1)	"an application library whose code space is (1) shared among all user mode applications, and (2) different than that occupied by the kernel and its associated files"
"A system, comprising . . . wherein the one or more isolated environments are created during installation of the one or more applications . . . where in the one or more isolated environments are removed as part of an uninstall of the one or more applications" ('500, '634, and '038 Patents, Claim 1)	**"wherein the one or more isolated environments are created during installation of the one or more applications":** "wherein the system is configured to create the one or more isolated environments during installation of the one or more applications" * * * **"wherein the one or more isolated environments are removed as part of an uninstall of the one or more applications":** "wherein the system is configured to remove the one or more isolated environments as part of an uninstall of the one or more applications"
"the system resources" ('500 Patent, Claim 19; '038 Patent, Claim 19)	"all of the system resources"
"appropriate for infrastructure configuration mapping" ('858 Patent, Claims 1, 19)	Plain and ordinary meaning, which requires an objective determination.
"instance of an image" ('858 Patent, Claims 1, 19)	"a physical or virtual occurrence of an operating system based on a file representation of a different instance"
"capturing" ('858 Patent, Claim 1)	Plain and ordinary meaning

"non-functional requirement" ('858 Patent, Claims 3-5 and 7)	Plain and ordinary meaning
"cloud infrastructure" ('858 Patent, Claims 1, 4–6, 8–12, 19)	"collection of hardware and software that enables on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service"

The Court **ORDERS** each party not to refer, directly or indirectly, to its own or any other party's claim-construction positions in the presence of the jury. Likewise, the Court **ORDERS** the parties to refrain from mentioning any part of this opinion, other than the actual positions adopted by the Court, in the presence of the jury. Neither party may take a position before the jury that contradicts the Court's reasoning in this opinion. Any reference to claim construction proceedings is limited to informing the jury of the positions adopted by the Court.

So ORDERED and SIGNED this 12th day of August, 2025.

RODNEY GILSTRAP
UNITED STATES DISTRICT JUDGE